UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/814,620	03/22/2001	Scott Patrick Hanson	ROC920000286US1	3982

7590 03/25/2005

Gero G. McClellan
Thomason, Moser & Patterson, L.L.P.
3040 Post Oak Boulevard, Suite 1500
Houston, TX 77056-6582

EXAMINER	
NAHAR, QAMRUN	

ART UNIT	PAPER NUMBER
2191	

DATE MAILED: 03/25/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/814,620	HANSON ET AL.	
	Examiner	Art Unit	
	Qamrun Nahar	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 January 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *12 January 2005* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the amendment filed on 1/12/05.

2. The objection to the specification is withdrawn in view of applicant's amendment.

3. Claims 1, 2 and 5 have been amended.

4. Claim 29 has been added.

5. Claims 1-29 are pending.

6. Claims 1-29 stand finally rejected under 35 U.S.C. 102(e) as being anticipated by Ungar (U.S. 6,085,035).

Response to Amendment

Drawings

7. The drawings were received on 1/12/05. These drawings are acceptable.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

9. Claims 1-29 are rejected under 35 U.S.C. 102(e) as being anticipated by Ungar (U.S. 6,085,035).

Per Claim 1 (Amended):

The Ungar patent discloses:

- **a method for optimizing a run time for an object code generated from a source code** ("The invention determines the type usage pattern for the data-values stored in the variable and accordingly optimizes some of the computer instructions used to access the variable." in column 3, lines 30-33)

- **extracting information for each procedure call contained in the source code** ("With a static compiler, the program is first profiled and the optimization process 300 is invoked during a subsequent compilation that utilizes the profiled data. The 'maintain type identifier' procedure 303 saves the type of a data-value stored in a variable in a type identifier associated with the variable. For example, the type identifier indicates when an integer data-value is stored in the variable. One skilled in the art will understand that some implementations of the invention use the type identifier to distinguish between a pointer and an integer. Other implementations use the type identifier to distinguish between a larger set of types (such as the pointer type and primary types). Next, a 'determine type usage pattern' procedure 305 evaluates the type mutability of the typed data-values stored in the variable. That is, the 'determine type usage pattern' procedure 305 determines whether only data-values having a specific type are stored in the variable. The typed data-values are type mutable if data-values of different types can be stored in the variable. However, if the stored data-values are of only one type the data-values are type immutable. This determination can be made from run-time data gathered by a profiler-

instrumented program compiled by a static compiler or by a the run-time and compiler states of a program compiled by a dynamic compiler. The 'determine type usage pattern' procedure 305 also determines, for mutable data-values, which types are most used (the preferred types)." in column 8, lines 11-41)

- **selecting a call linkage between a caller procedure and a callee procedure for each procedure call using the extracted information, where the selected call linkage is optimized to minimize the run time of the object code generated from the source code; generating the object code from the source code; and running the object code using the selected call linkages for each procedure call** ("A first preferred embodiment optimizes both a called routine and the call site dependent on the types of the data-value passed from the call site to the called routine. Because data-values contained in passed entities (that is, the data-values contained in variables and/or the addresses of the variables themselves) can be specified as arguments to, or a result from, a called routine, the call site generally contains code to select which executable version of the called routine to invoke dependent on the types of the passed entities. The invention detects variables that have immutable types (from the 'determine type usage pattern' procedure 305) and optimizes both the called routine and the call site dependent upon the type-mutability of the passed entities. Additionally, if the variables have mutable types, the invention generates multiple versions of the called routine (each optimized for a preferred type as determined by the 'determine type usage pattern' procedure 305) that are invoked dependent on the types of the passed data-values. Often, one of these called routine versions is not optimized

with respect to any of the passed data-values and so is capable of processing any pattern of types of the passed data-value." in column 8, lines 52-67 to column 9, lines 1-7).

Per Claim 2 (Amended):

The Ungar patent discloses:

- wherein the selected call linkage is one of a memory-based call linkage and a register-based call linkage (column 9, lines 22-67 to column10, lines 1-21).

Per Claim 3:

The Ungar patent discloses:

- wherein, if the memory-based call linkage is selected for a particular procedure call, the running comprises: allocating a block in a memory to store a value for each argument in the particular procedure call; storing the value for each argument from a register in a processor to the block in the memory; branching the procedure call to a callee procedure, and loading the value for each argument from the block in the memory back to the register (column 10, lines 7-21).

Per Claim 4:

The Ungar patent discloses:

- wherein, if the register-based call linkage is selected for a particular procedure call, the running comprises: copying a value, for each argument in a procedure call, from a register in the processor to a parameter register in the processor; branching the procedure call to a callee procedure; and copying the value from the parameter register back to the register (column 9, lines 60-65).

Per Claim 5 (Amended):

The Ungar patent discloses:

- wherein the selecting comprises: detecting whether an error exists for the procedure call; selecting the memory based call linkage if the error is detected for the procedure call; and selecting the register-based call linkage if no error is detected for the procedure call (column 9, lines 60-65 and column 10, lines 7-21).

Per Claim 6:

The Ungar patent discloses:

- wherein the error is detected if the procedure call has a different number of parameters than the callee procedure (column 10, lines 7-21).

Per Claim 7:

The Ungar patent discloses:

- wherein the error is detected if a parameter type for a parameter in the caller procedure is different than the parameter type for the parameter at a corresponding position in the callee procedure (column 10, lines 7-21).

Per Claim 8:

The Ungar patent discloses:

- wherein the error is detected if a number of arguments in the procedure call is greater than a number of parameter registers used to run the object code (column 10, lines 22-50).

Per Claim 9:

The Ungar patent discloses:

- wherein the error is detected if an argument in the procedure call is unpassable in a register (column 10, lines 51-61).

Per Claim 10:

The Ungar patent discloses:

- **wherein the extracting of procedure call information comprises: extracting information for each procedure definition contained in the source code** (column 8, lines 11-41 and column 9, lines 8-21).

Per Claim 11:

The Ungar patent discloses:

- **wherein the extracted procedure call information comprises an identifier for a calling procedure and a callee procedure, and the extracted procedure definition information comprises a number of arguments received by the procedure and a classification for each argument** (column 9, lines 35-67 to column 10, lines 1-21).

Per Claim 12:

The Ungar patent discloses:

- **wherein the call linkage is selected in a class comprising one of a register stacks call linkage, a system call linkage and a near versus far call linkage** (column 9, lines 22-65).

Per Claim 13:

The Ungar patent discloses:

- wherein the extracted information is generated in a data structure used to select the call linkage for each procedure call (column 9, lines 8-21).

Per Claims 14-15:

These are apparatus versions of the claimed method discussed above, claim 1, wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above, including "a memory for storing a compiler program; and a processor comprising a plurality of registers, where a subset of the plurality of registers comprise parameter registers" (column 3, lines 44-57). Thus, accordingly, these claims are also anticipated by Ungar.

Per Claims 16-20:

These are apparatus versions of the claimed method discussed above (claims 2, 5, 10, 11 and 13, respectively), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Ungar.

Per Claims 21-28:

These are computer readable medium versions of the claimed method discussed above (claims 1-5, 10-11 and 13, respectively), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Ungar.

Per Claim 29 (New):

This is a computer readable medium version of the claimed method discussed above, claim 1, wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above, including "(iii) linking the object code modules according to the selected call linkage, and (iv) generating the executable program from the object code modules" (column 7, lines 50-62; and column 11, lines 66-67 to column 13, lines 1-13). Thus, accordingly, this claim is also anticipated by Ungar.

Response to Arguments

10. Applicant's arguments filed on 1/12/05 have been fully considered but they are not persuasive.

In the remarks, the applicant argues that:

a) Claims 1-28 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Respectfully, applicants traverse the objection. Ungar and Ungar (US 6,085,035). Applicants' invention do both involve complier optimization techniques', however, Ungar does not teach what is presently claimed. To understand the differences a brief discussion about compiler functioning is warranted.

A compiler is a computer program that translates a computer program ...

...

Ungar discloses a method for optimizing the object code generated by a compiler using the data types associated with variables. In particular, the optimization techniques of Ungar describes variable types as "mutable" and "immutable" to define meta-information about a particular variable to optimize object code.

Applicants claim a method for optimizing call-linkage for each call to a procedure that occurs in a program, regardless of whether a variable is "mutable" or "immutable." In contrast, as described below, Ungar discloses a complier optimization technique using the data types associated with a computer program's variables. Put simply, although both Applicants' claims and Ungar may be used to optimize object code generated from source modules, they do so in different ways, using different information, to achieve different optimization results.

Regarding claims 1-13: Applicants claim "a method for optimizing a run time for an object code generated from a source code" that includes the steps of (i) extracting information for each procedure call contained in the source code, (ii) linkage between a caller procedure and a callee procedure for each selecting a call procedure call using the extracted information, where the selected call linkage is optimized to minimize a run time of an object code generated from the source code, (iii) generating the object code from the source code, and (iv) running the object code using the selected call linkages for each procedure call. Admittedly, both Ungar and Applicants' claims are directed to optimization methods to optimize object generated from source code, and both include generating the object code from the source code (i.e., both contemplate actually compiling the source code to produce an executable computer program). Beyond this, however, the similarities end.

As stated, Applicants claim an optimization technique that includes (i) extracting information for each procedure call contained in the source code, (ii) selecting a call linkage between a caller procedure and a called procedure for each procedure call using the extracted information, where the selected call linkage is optimized to minimize a run time of an object code generated from the source code. Ungar teaches a method for optimizing object code based

on variable "types." Ungar Col. 3 1. 30-35. As used by Ungar, a variable may be "immutable" or "mutable" depending on whether a variable can (or actually does) store different data types. Ungar Col. 8 1. 1 1-52, Figure 3. Using a dynamic compilation environment (i.e., the compiler runs concurrent with program execution) the method taught by Ungar monitors the executing program to determine actual, or likely variable types and usage patterns, and creates compiled object code accordingly. Ungar Col. 8 1. 35-40. Ungar also teaches that in a non-dynamic environment "using a static compiler, the program is first profiled and the optimization process is invoked during a subsequent compilation that utilizes the profiled data." Ibid. In either case, the methods disclosed by Ungar require data regarding program execution. This data may be obtained from an executing program in a dynamic compilation environment, or from a monitoring process that monitors the actively of an executable program compiled in a static compilation environment.

The present claims recite optimizing source code by selecting on one of multiple call linkage options determined using information extracted from the source code for the program being compiled contained in multiple modules. In contrast, Ungar discloses optimizing executable code based on profiling data obtained either from an executing program used for a subsequent compilation (and optimization) of the same source program or from monitoring an executing program and compiling subsequent portions of the program based on the monitored activity. For example, the Ungar discloses: "A first preferred embodiment optimizes both a called routine and the call site dependent on the types of data values passed from the call site to the called routine. ...The invention detects variables that have immutable types (from the 'determine type usage pattern' 305)." Ungar, Col. 8 1. 52-67. The "determine type usage pattern" disclosed

by Ungar requires the compiler to determine executing program behavior to optimize the executable code. "This determination can be made from run-time data gathered by a profiler-instrumented program compiled by a static compiler or by a [sic] the run-time and compiler states of a program compiled by a dynamic compiler." Ungar, Col. 8 1. 35- 40.

Applicants, however, claim an optimization technique that relies on selecting a call linkage between a caller procedure and a callee procedure for each procedure call using the extracted information, where the selected call linkage is optimized to minimize a run time of an object code generated from the source code. The extracted information is derived from the various source modules being compiled by the complier not from the "type usage patterns" taught by Ungar. Applicants' claim selecting a call linkage based only on the information extracted from the source code, without any reliance (or even awareness) of later program execution, whereas Ungar discloses an optimization technique that requires some reliance on executing program behavior, Ungar, Col. 8 1. 35-40.

Accordingly, Ungar fails to teach show or suggest a method for optimizing a run time for an object code generated from a source code that includes the steps of extracting information for each procedure call contained in the source code, selecting a call linkage between a caller procedure and a callee procedure for each procedure call using the extracted information, where the selected call linkage is optimized to minimize the run time of the object code generated from the source code, and generating the object code from the source code; and running the object code using the selected call linkages for each procedure call.

Examiner's response:

a) Examiner strongly disagrees with applicant's assertion that Ungar fails to disclose the claimed limitations recited in claims 1-13. Ungar clearly shows each and every limitation in claims 1-13.

Ungar teaches extracting information for each procedure call contained in the source code (column 8, lines 11-41), where information is extracted using run time data. Claim 1 does not recite **when and how** the step of extracting information is done. Therefore, in response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., "The extracted information is derived from the various source modules being compiled by the complier") are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

In addition, see the rejection above in paragraph 9 for rejection to claims 1-13.

In the remarks, the applicant argues that:

b) Regarding claims 14-28: Claims 14-20 claim an apparatus for optimizing a run time of an object code generated from a source code, and claims 21-28 claim an a computer readable medium storing a software program that performs the operations of Applicants' invention. The Examiner rejects these claims as "all claim limitations also have been addressed and/or covered in the cited areas as set forth above," and the rejection of these claims does not assert any independent basis to support the rejection. Accordingly, the traversal to the rejection of claims 1-13 as set forth above applies with equal force to claims 14-28.

The secondary references made of record are noted. However, it is believed that the secondary references are no more pertinent to the Applicants' disclosure than the primary references cited in the office action. Therefore, Applicants believe that a detailed discussion of the secondary references is not necessary for a full and complete response to this office action.

Examiner's response:

b) The Examiner has already addressed the applicant's arguments regarding claims 1-13 in the Examiner's Response (a) above. In addition, see the rejection above in paragraph 9 for rejection to claims 14-28.

Conclusion

11. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

12. Any inquiry concerning this communication from the examiner should be directed to Qamrun Nahar whose telephone number is (571) 272-3730. The examiner can normally be reached on Mondays through Thursdays from 8:30 AM to 6:00 PM. The examiner can also be reached on alternate Fridays.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki, can be reached on (571) 272-3719. The fax phone number for the organization where this application or processing is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

QN
March 18, 2005

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100